

**NAME**

curl\_easy\_getinfo - extract information from a curl handle

**SYNOPSIS**

```
#include <curl/curl.h>
```

```
CURLcode curl_easy_getinfo(CURL *curl, CURLINFO info, ... );
```

**DESCRIPTION**

Request internal information from the curl session with this function. The third argument **MUST** be a pointer to a long, a pointer to a char \*, a pointer to a struct curl\_slist \* or a pointer to a double (as this documentation describes further down). The data pointed-to will be filled in accordingly and can be relied upon only if the function returns CURLE\_OK. This function is intended to get used AFTER a performed transfer, all results from this function are undefined until the transfer is completed.

You should not free the memory returned by this function unless it is explicitly mentioned below.

**AVAILABLE INFORMATION**

The following information can be extracted:

**CURLINFO\_EFFECTIVE\_URL**

Pass a pointer to a 'char \*' to receive the last used effective URL.

**CURLINFO\_RESPONSE\_CODE**

Pass a pointer to a long to receive the last received HTTP or FTP code. This option was known as CURLINFO\_HTTP\_CODE in libcurl 7.10.7 and earlier.

**CURLINFO\_FILETIME**

Pass a pointer to a long to receive the remote time of the retrieved document. If you get -1, it can be because of many reasons (unknown, the server hides it or the server doesn't support the command that tells document time etc) and the time of the document is unknown. Note that you must tell the server to collect this information before the transfer is made, by using the CURLOPT\_FILETIME option to *curl\_easy\_setopt(3)*. (Added in 7.5)

**CURLINFO\_TOTAL\_TIME**

Pass a pointer to a double to receive the total transaction time in seconds for the previous transfer. This time does not include the connect time, so if you want the complete operation time, you should add the CURLINFO\_CONNECT\_TIME.

**CURLINFO\_NAMELOOKUP\_TIME**

Pass a pointer to a double to receive the time, in seconds, it took from the start until the name resolving was completed.

**CURLINFO\_CONNECT\_TIME**

Pass a pointer to a double to receive the time, in seconds, it took from the start until the connect to the remote host (or proxy) was completed.

**CURLINFO\_PRETRANSFER\_TIME**

Pass a pointer to a double to receive the time, in seconds, it took from the start until the file transfer is just about to begin. This includes all pre-transfer commands and negotiations that are specific to the particular protocol(s) involved.

**CURLINFO\_STARTTRANSFER\_TIME**

Pass a pointer to a double to receive the time, in seconds, it took from the start until the first byte is just about to be transferred. This includes CURLINFO\_PRETRANSFER\_TIME and also the time the server needs to calculate the result.

**CURLINFO\_REDIRECT\_TIME**

Pass a pointer to a double to receive the total time, in seconds, it took for all redirection steps include name lookup, connect, pretransfer and transfer before final transaction was started. CURLINFO\_REDIRECT\_TIME contains the complete execution time for multiple redirections.

(Added in 7.9.7)

#### CURLINFO\_REDIRECT\_COUNT

Pass a pointer to a long to receive the total number of redirections that were actually followed.

(Added in 7.9.7)

#### CURLINFO\_SIZE\_UPLOAD

Pass a pointer to a double to receive the total amount of bytes that were uploaded.

#### CURLINFO\_SIZE\_DOWNLOAD

Pass a pointer to a double to receive the total amount of bytes that were downloaded. The amount is only for the latest transfer and will be reset again for each new transfer.

#### CURLINFO\_SPEED\_DOWNLOAD

Pass a pointer to a double to receive the average download speed that curl measured for the complete download.

#### CURLINFO\_SPEED\_UPLOAD

Pass a pointer to a double to receive the average upload speed that curl measured for the complete upload.

#### CURLINFO\_HEADER\_SIZE

Pass a pointer to a long to receive the total size of all the headers received.

#### CURLINFO\_REQUEST\_SIZE

Pass a pointer to a long to receive the total size of the issued requests. This is so far only for HTTP requests. Note that this may be more than one request if FOLLOWLOCATION is true.

#### CURLINFO\_SSL\_VERIFYRESULT

Pass a pointer to a long to receive the result of the certification verification that was requested (using the CURLOPT\_SSL\_VERIFYPEER option to *curl\_easy\_setopt(3)*).

#### CURLINFO\_SSL\_ENGINES

Pass the address of a 'struct curl\_slist \*' to receive a linked-list of OpenSSL crypto-engines supported. Note that engines are normally implemented in separate dynamic libraries. Hence not all the returned engines may be available at run-time. **NOTE:** you must call *curl\_slist\_free\_all(3)* on the list pointer once you're done with it, as libcurl will not free the data for you. (Added in 7.12.3)

#### CURLINFO\_CONTENT\_LENGTH\_DOWNLOAD

Pass a pointer to a double to receive the content-length of the download. This is the value read from the Content-Length: field.

#### CURLINFO\_CONTENT\_LENGTH\_UPLOAD

Pass a pointer to a double to receive the specified size of the upload.

#### CURLINFO\_CONTENT\_TYPE

Pass a pointer to a 'char \*' to receive the content-type of the downloaded object. This is the value read from the Content-Type: field. If you get NULL, it means that the server didn't send a valid Content-Type header or that the protocol used doesn't support this.

#### CURLINFO\_PRIVATE

Pass a pointer to a 'char \*' to receive the pointer to the private data associated with the curl handle (set with the CURLOPT\_PRIVATE option to *curl\_easy\_setopt(3)*). (Added in 7.10.3)

#### CURLINFO\_HTTPAUTH\_AVAIL

Pass a pointer to a long to receive a bitmask indicating the authentication method(s) available. The meaning of the bits is explained in the CURLOPT\_HTTPAUTH option for *curl\_easy\_setopt(3)*. (Added in 7.10.8)

#### CURLINFO\_PROXYAUTH\_AVAIL

Pass a pointer to a long to receive a bitmask indicating the authentication method(s) available for your proxy authentication. (Added in 7.10.8)

**CURLINFO\_OS\_ERRNO**

Pass a pointer to a long to receive the errno variable from a connect failure. (Added in 7.12.2)

**CURLINFO\_NUM\_CONNECTS**

Pass a pointer to a long to receive how many new connections libcurl had to create to achieve the previous transfer (only the successful connects are counted). Combined with *CURLINFO\_REDIRECT\_COUNT* you are able to know how many times libcurl successfully reused existing connection(s) or not. See the Connection Options of *curl\_easy\_setopt(3)* to see how libcurl tries to make persistent connections to save time. (Added in 7.12.3)

**RETURN VALUE**

If the operation was successful, CURLE\_OK is returned. Otherwise an appropriate error code will be returned.

**SEE ALSO**

**curl\_easy\_setopt(3)**